

**Konvertor HTML stránek do formátu  
elektronických knih určený pro mobilní  
zařízení**

**HTML to eBook Format Converter for  
Mobile Devices**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2010

.....

## Abstrakt

Tato práce se zabývá realizací konvertoru HTML stránek do formátu elektronických knih. Implementace vychází ze známých formátů elektronických knih a jejich obsahových možností, jako je například formátování textu či vkládání obrázků. Obsahuje nejen srovnání jednotlivých formátů, ale také programů určených pro jejich četbu. Zdůvodňuje se zde i výběr programovacího prostředí, jehož bude pro implementaci využito. Výsledkem je konvertor převádějící HTML stránky do dvou různých formátů. Kvalitou převodu a převedených dokumentů odpovídající stávajícímu programu sloužícímu k témuž účelu, ale pracujícímu na běžném počítači.

**Klíčová slova:** JavaME, MIDP, CLDC, konvertor, čtečka, mobilní telefon, elektronická kniha, ePub, fb2, PRC, MobiPocket, TXT, Plucker, iSilo

## Abstract

This thesis deals with the implementation of converter from HTML pages to ebooks. Implementation is based on popular formats of electronic books and their available parameters, such as text formatting and image insertion. It contains comparison of different formats and of programs designed for e-Book reading. The choice of implementation environment is also justified. The result is a converter, transforming HTML pages into two different e-Book formats. The resulting conversion quality is consistent with existing converters.

**Keywords:** JavaME, MIDP, CLDC, converter, reader, cell phone, e-book, ePub, fb2, PRC, MobiPocket, TXT, Plucker, iSilo

## Seznam použitých zkratk a symbolů

CF	- Compact Framework
CLDC	- Connected Limited Device Configuration
ePub	- Electronic Publication
FB2	- Fiction Book version 2
HTML	- Hyper Text Markup Language
JavaME	- Java Micro Edition
JavaSE	- Java Standard Edition
MIDP	- Mobile Information Device Profile
PDA	- Personal Digital Assistant
PDB	- Palm Database
TXT	- Text File
XML	- Extensible Markup Language

## Obsah

1	Úvod.....	1
2	Elektronické knihy .....	2
2.1	Formáty Elektronických knih .....	3
2.1.1	PalmDoc (PDB).....	3
2.1.2	Mobipocket (PRC).....	4
2.1.3	FictionBook 2 (FB2).....	5
2.1.4	Open E-Book (OEB) .....	6
2.1.5	EPUB .....	6
2.1.6	Plucker .....	7
2.1.7	iSilo™.....	7
2.1.8	Text (TXT) .....	8
2.1.9	Srovnání jednotlivých formátů .....	9
2.2	Čtečky elektronických knih .....	10
2.2.1	FBReader a FBReaderJ .....	10
2.2.2	Haali Reader .....	11
2.2.3	ReadManiac .....	11
2.2.4	MobiPocket Reader .....	12
2.2.5	iSilo.....	12
2.3	Konvertory elektronických knih.....	12
3	Implementační prostředí pro vývoj konvertoru.....	14
3.1	Základní konfigurace .....	14
3.2	Nestandardní knihovny.....	14
4	Analýza programu .....	16
4.1	Základní popis programu.....	16
4.2	Uživatelské rozhraní .....	17
4.3	Převod formátů .....	18
5	Implementace programu.....	20
5.1	Grafické rozhraní .....	20
5.2	Příprava převodu .....	20
5.3	Realizace převodu.....	22
5.4	Přidání nového formátu .....	23
6	Testování programu .....	25
7	Závěr .....	28
8	Literatura .....	29
9	Seznam příloh.....	30
A	Uživatelská příručka.....	31

# 1 Úvod

V dnešní době se již kromě obyčejných tištěných knih rozvíjí i knihy elektronické. Pro tyto knihy neexistuje jednotný formát, ale celá řada. Ke čtení takovýchto knih je možno využít zařízení konstruovaných jen pro tento účel nebo využít stávajících technických prostředků, jako jsou například počítače. To nám však nepřináší patřičnou mobilitu. Proto vznikly různé čtecí programy umožňující nám číst elektronické knihy i na mobilních zařízeních. Jejich hlavní funkcí je námi dodanou knihu přečíst. Pokud potřebujeme takovouto knihu vytvořit, musíme tak převážně učinit opět na počítači. Z toho důvodu bude vytvořen konvertor HTML stránek do vybraných formátů elektronických knih.

Následující kapitola bude zaměřena na samotné elektronické knihy a na jejich možnosti. Jelikož se vyskytují v mnoha podobách (formátech) s různými vlastnostmi, budou probrány některé z nejrozšířenějších formátů. Dále bude kapitola zaměřena na možnost jejich čtení a samotné čtecí programy implementované převážně pro mobilní zařízení. V jejím závěru bude zmíněna možnost převodu jiných formátů právě na elektronické knihy.

Třetí kapitola pojednává o výběru implementačního prostředí pro konvertor, jenž je cílem této práce. Je v ní zmíněn a odůvodněn výběr implementačního prostředí.

Další kapitola se zabývá analýzou konvertoru. Jeho výslednými požadavky a vlastnostmi. Také pojednává o možnostech řešení základních problémů spojených s budoucí implementací tohoto programu.

V páté kapitole této práce je již zmíněna samotná implementace. Hlavní důraz je kladen na její základní prvky. Obsahuje části věnované grafickému rozhraní, třídě připravující potřebné věci na převod a pak samotné části, jež tento převod realizuje.

Poslední kapitola je založena na výsledném programu. Posuzuje implementovaný program a jeho výstupní soubory. Také srovnává program s jiným, existujícím pro běžné počítače. Jedná se o program Any to FB2. Jsou srovnány výstupní soubory z obou programů a také doba a kvalita převodu.

## 2 Elektronické knihy

Elektronická kniha je obdoba běžné tištěné knihy, z níž může být převedena nebo přímo elektronicky vytvořena autorem. Někdy jsou však tak nazývány i zařízení pro čtení takovýchto knih. Ty jsou však mnohem méně rozšířené, než samotné užívání elektronických knih.

Elektronické publikování je pro autory mnohdy výhodnější. Mají nad publikací svého díla mnohem větší přehled a také se snižují finanční náklady spojené s běžným publikováním. Ne vždy se však takováto varianta setkává s porozuměním čtenářů. Pro většinu z nich je při čtení dlouhých textů příjemnější sledovat běžnou tištěnou knihu než relativně malý displej elektronického zařízení určeného pro čtení knihy. Je však nutno dbát i toho, že toto zařízení si zachová svou velikost, i když v něm bude nahráno více knih, zatímco v tištěné podobě by stejný počet knih již značně zvýšil jejich celkovou váhu a rozměry. Elektronické knihy lze samozřejmě pomocí čtecích programů číst i na každém běžném počítači. Ten se však svými rozměry a pohodlím spojeným se čtením knih jeho prostřednictvím stává mírně nevyhovujícím.

Ať už by se jednalo o knihu tištěnou nebo technické zařízení pro čtení elektronických knih, je to vždy něco navíc, co sebou musí čtenář nést, aby měl možnost číst kdykoli a kdekoli. Z toho důvod se začaly vyvíjet i programy, umožňující číst knihy prostřednictvím něčeho, co má u sebe v dnešní době většina lidí. Jedná se běžné mobilní telefony a PDA.

Programy umožňující čtení elektronických knih však nejsou jednotné a podporují různé formáty, což je mnohdy spojeno s podobou samotného zařízení, pro něž je daný program určen. I při stejných formátech stále může docházet k rozdílům. Jednotlivé programy vytvářejí různí vývojáři, či týmy vývojářů. Tím vznikají i odlišnosti v podporovaných funkcích samotných knih stejného formátu. Některé vlastnosti formátů nemusí být podporovány nebo se zobrazují naprosto jinak než v dalším programu. U běžné elektronické knihy to není na závadu a většinou to čtení knihy ani nijak neomezí. Může však docházet ke snížení orientace v dané knize. Jako příklad může sloužit situace, kdy formát umožňuje zobrazování obsahu či používání odkazů, zatímco program sice daný formát podporovat bude, ale nemá vnitřní implementaci právě pro tyto dva zmíněné prvky.

Také je třeba dbát na to, že čtecí programy jsou implementovány vždy implementovány vždy v nějakém programovacím jazyce. Většinou se jedná o JavaME, nebo .NET CF. Běžné mobilní telefony podporují pouze první z nich, zatímco s druhým si neporadí, protože tato platforma není v telefonu obsažena. Při výběru takového programu se tedy musí čtenář rozhodovat první právě na základě toho, jaké zařízení vlastní, či jakou platformu toto zařízení obsahuje. Dalším kritériem jsou podporované formáty a následně právě jejich podporované vlastnosti.

## 2.1 Formáty Elektronických knih

### 2.1.1 PalmDoc (PDB)

PalmDoc je jeden z nejstarších formátů určených pro čtení v mobilních zařízeních. Konkrétně byl určen pro použití v PDA s operačním systémem PalmOS. Je to však pouze původní účel. Pro svou jednoduchost je tento formát čitelný i na mnoha jiných zařízeních. Kniha ve formátu PDB může klidně obsahovat pouze čistý text. Základní verze PDB neumožňuje žádné formátování textu ani přidávání jakýchkoli jiných prvků kromě textu. Existují však již i další jeho rozšíření, kde lze na základě různých značek (způsob lze přirovnat k HTML) přidávat další prvky do PDB souboru. Zde však nastává problém se vzájemnou kompatibilitou. Každý si totiž díky tomu, že PDB je otevřený formát, může doimplementovat vlastní rozšíření.

Velkou nevýhodou PDB je nemožnost využívání znaků z nových typů kódování Unicode. Pro češtinu jsou pouze dvě možnosti, ISO-8859-2 a Windows-1250. Převážná většina knih je však vytvořena v kódování Windows-1250.

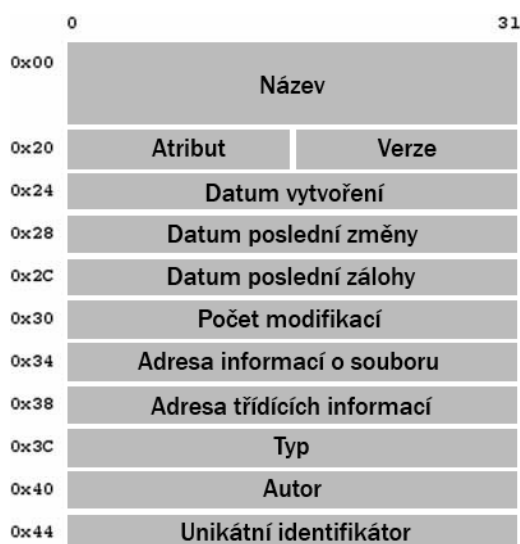
PDB je znám i svou velikou kompresí (umožněna díky základní čistě textové verzi). Pokud se vytvoří jeden textový soubor (např. kniha), tak tato výhoda nebude příliš uplatněna, protože komprese bude téměř nulová. Avšak pokud je zapotřebí uchovávat nebo posílat obrovské množství dat, je už výhodné využít tento formát. Existuje na něj spousta čtecích programů a je možno jej vytvořit i ve freeware editorech (např. PSPad).

<b>Záhlaví</b>
<b>Seznam záznamů</b>
<b>Informace o aplikaci</b>
<b>Třídící informace</b>
<b>Data</b>

Obrázek 1: Struktura PDB souboru



PDB soubor má svou pevnou strukturu. Jako první je v něm zapsáno záhlaví (PDB header). To obsahuje veškerá metadata o souboru (viz. obrázek 2). Následuje Seznam záznamů (Record list), aplikační a třídící informace. Jako poslední je v souboru umístěn samotný záznam (Record), což je vlastní obsah souboru. Celá struktura je znázorněna na obrázku 1.



Obrázek 2: PDB hlavička

### 2.1.2 Mobipocket (PRC)

Formát PRC je již značně pokročilejší než PDB. Podporuje spoustu klasických prvků formátování a ještě mnoho dalších. Mezi ty klasické patří veškeré formátování samotného písma, typy (kurzíva, tučné, podtržené), velikost. Je možno specifikovat nadpisy a kapitoly. Dokonce lze i vytvářet obsahy, záložky, přidávat různé poznámky a náčrty. Má již plnou podporu pro hypertextové odkazy.

Mobipocket má ještě hodně dalších možností rozšíření, ale ty jsou závislé na platformě. Plnou podporu všech prvků formátu Mobipocket zaručuje pouze MS Windows. S ním je možno vytvářet i takové dokumenty jako jsou různé kvízy, slovníky či jednoduché databáze. Při použití jiné platformy se obsah dokumentu nijak nemění, pouze jsou zanedbány funkce, jež nejsou na daném zařízení podporovány.

Struktura PRC souboru je téměř stejná jako u PDB. Jednotlivé části se však mohou vyskytovat pod odlišnými označeními.

### 2.1.3 FictionBook 2 (FB2)

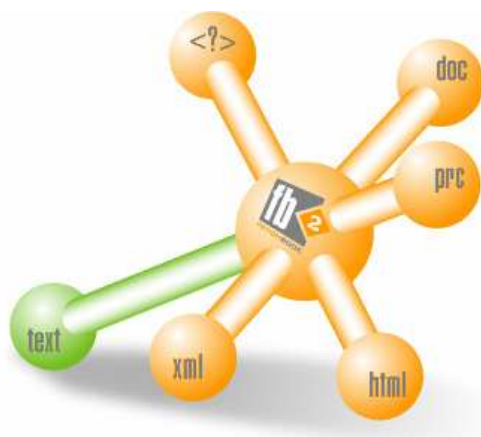
Tento formát pochází z Ruska, kde si i získal velkou oblibu. Vychází z již zavedené struktury XML formátu. Samotný vzhled textu nespecifikuje, avšak zaměřuje se na formátování a logickou strukturu celého dokumentu. Podporuje vytváření odstavců, kapitol a jejich nadpisů. Pro jednotlivé podporované výrazy má přednastaven vlastní vzhled. Kromě základních struktur definuje také mnoho dalšího, co se objevuje v běžných knihách. Mezi takovéto prvky, jež jsou zároveň specifikovány v FB2 formátu, patří například poznámky autora, verše a citace. Od verze 2 jsou implementovány i hypertextové odkazy. Přestože jsou založeny na XML, tak jsou zde použitelné jen omezeně a to v rámci jednoho souboru. Jakýkoli odkaz na jiný soubor (knihu), není možný. Formát FB2 umožňuje vkládat do dokumentu ilustrace.

Formát FB2 se podobně jako kterýkoli HTML dokument skládá z hlavičky (*head*) a těla (*body*). V hlavičce mohou být uvedeny všechny myslitelné údaje o dané knize. Patří zde takové údaje jako žánr knihy, její autor, název, datum vydání, program využitý k vytvoření její elektronické podoby a také datum kdy se tak stalo. Je možno knize přiřadit i stylový CSS (Cascading Style Sheets) soubor. Jeden dokument obsahuje pouze jednu sekci *head*. S *body* už to tak jednoznačné není. V implementaci knihy můžeme nalézt více těchto sekcí. Využívá se toho k oddělení částí přímo nesouvisejících s vlastním textem knihy. Například pro různé poznámky či vysvětlivky. Dále se v tělu dokumentu využívá rozdělování na sekce, přičemž každá část *body* musí mít alespoň jednu sekci. Ty je možno dále rozdělit a to na jednotlivé odstavce, je to však pouze možnost, ne nutnost.

```
<description>
  <title-info>
    <genre> Žánr </genre>
    <author>
      <first-name> Jméno </first-name>
      <last-name> Příjmení </last-name>
    </author>
    <book-title> Název knihy </book-title>
    <annotation> </annotation>
    <date>2009</date>
  </title-info>
  <document-info>
    <author>
      <first-name> <first-name/>
      <last-name> <last-name/>
      <nickname> </nickname>
    </author>
    <program-used> Použitý program </program-used>
    <date> 2009-23-01 </date>
    <src-ocr> </src-ocr>
    <version> 1.2 </version>
    <history> </history>
  </document-info>
</description>
```

Obrázek 3: FB2 - ukázka hlavičky souboru

Tento formát je velice snadno konvertovatelný do mnoha jiných formátů (obrázek 1). Je však možné, že jsme přidali našeho FB2 dokumentu nějaký vlastní styl. Pak je nutno při konverzi do jiných formátů tento soubor také přiložit. Pokud tedy předem víme, že naše kniha bude časem dále převáděna i do jiných formátů, je jistější držet se pouze již implementovaných stylů, které by nám měly pro normální elektronickou knihu vystačit. Vyhneme se tak případným dalším problémům. Pokud bychom konvertovali do obyčejného textového souboru, stejně by se žádné přidání stylů nevyužily.



Obrázek 4: konvertovatelnost formátů do FB2  
Převzato z: [5]

#### 2.1.4 Open E-Book (OEB)

OEB je otevřený formát založený na XHTML. Podporuje samozřejmě také kaskádové styly (CSS). Celá se skládá ze ZIP balíčku a manifestu, v němž jsou obsažena metadata samotné elektronické knihy. V dnešní době se však již téměř nevyužívá, protože ho nahradil velice podobný formát s více možnostmi EPUB.

#### 2.1.5 EPUB

Tento formát je nástupcem OEB. Shodně také vychází z XHTML. Je zde však již podpora vyšší verze CSS (CSS2). Díky stylům je možno libovolně formátovat písmo i celý text, odstavce, obrázky.

Veškerá metadata jsou obsažena v několika XML souborech. Ty jsou následně sloučeny do jednoho archívu standardní ZIP technologií. Archivační technologie ZIP je velice rozšířená, takže se setkává s širokou podporou ze strany čtecích zařízení. Hlavní problém však nastává díky XHTML. Je sice velice výhodné, že můžeme využívat mnoho prvků a nástrojů jak pro vytváření, tak i pro prohlížení, ale je pak velice obtížné všechny tyto nástroje naprosto stejně naimplementovat do čtecích zařízení. Proto se může stát, že stejná kniha v tomto formátu může na různých zařízeních vypadat různě.

### 2.1.6 Plucker

Původně formát navržený pro offline čtení HTML stránek. Ty jsou první staženy a následně transformovány do plucker dokumentu. K tomu je využíváno většinou nástrojů třetích stran.

Již z jeho původního určení vyplývá, že tento formát je přizpůsoben svými dostupnými vlastnostmi používaným prvkům v HTML souborech. Podporuje formátování textu a také obrázky. Celý výsledný soubor je pouze komprimovaným HTML archivem v binární formě.

### 2.1.7 iSilo™

Tento formát je sice uzavřený, ale i přes tuto vlastnost velice rozšířený. Pod názvem iSilo™ je však více znám samotný stejnojmenný program pro čtení elektronických knih. I přes svůj název totožný se samotným formátem zvládne přečíst i některé další typy elektronických knih.

iSilo™ dokument má možnost zabezpečení proti neoprávněnému používání i zásahům do souboru. Jsou však dostupné až od verze iSilo™ 3.x. Lze ho chránit heslem, jež nám nedovolí provádět některé operace. Konkrétně zde patří kopírování celého souboru, kopírování jeho částí do schránky, modifikace, konverze do jiných formátů a tisk. Další možností, jež se týká zabezpečení, je vložení registračního kódu. Ten se vkládá pouze jednou a to při prvotním vytvoření souboru. Obsahuje mimo datum vytvoření také speciální identifikační kód autora. Tento formát zahrnuje ještě další bezpečnostní prvek. Do souboru lze ještě přidat datum expirace. Jakmile vyprší, tak již nelze dokument otevřít. Dle prohlášení tvůrců má toto opatření zabránit dalšímu kolování zastaralých dokumentů, jejichž platnost již vypršela.

Autoři iSila™ kladou důraz i na poměrně vysokou kompresi textu využitou v jejich formátu. Při převedení čistého textu do iSilo™ formátu dosáhne výsledný soubor 50-40% velikosti oproti souboru původnímu. To znamená 50-60% úsporu místa. Tvůrci formátu uvádějí, že iSilo™ má

o 20% lepší kompresi než samotný Palm Doc formát. Srovnání kompresí obou formátů je uvedeno v tabulce 1. Původní textový soubor je převeden do Palm Doc a do dvou verzí iSilo™.

Formát dokumentu	Velikost [byte]	Úspora [byte]	Redukce [%]
Text	3,282,452	-	-
Palm Doc	1,800,331	1,482,121	45.15%
iSilo™ 2.x	1,478,324	1,804,128	54.96%
iSilo™ 3.x	1,472,359	1,810,093	55.14%

Tabulka 1: Porovnání komprese formátu Palm Doc a iSilo™  
Převzato z: literatura [7]

Možnostmi, jež poskytuje iSilo™ v rámci formátování textu, by se dal tento formát přirovnat k HTML. Umožňuje totiž libovolné typy písma, jeho velikost, dokonce implementuje i horní a dolní index. Lze měnit barvu písma i jeho pozadí. Dovoluje vkládat libovolně velké obrázky, kvůli úspoře místa jsou opět komprimovány. Je možno je libovolně umístit v textu nebo mimo něj. Také systém týkající se tabulek je v tomto případě velice podobný tomu z HTML. Funguje nastavování šířky a výšky buněk či celé tabulky, také přesahování buněk do sousedních. Jakýkoli text i obrázek může sloužit jako odkaz na jinou část dokumentu. Jednou z výhod tohoto formátu je i vkládání a formátování celých bloků, je možno jim nastavit obtékání, zarovnání, barvu pozadí a orámování.

Pro konverzi do iSilo™ existuje oficiální aplikace iSiloX. Slouží však pouze pro převod HTML stránek do formátu iSilo™.

### 2.1.8 Text (TXT)

V případě standardního textového formátu se jedná pouze o čistý text. Není zde žádná možnost rozšířeného formátování kromě odstavců, konců řádků, tabulátorů a mezer. V některých čtečkách však existuje jistá vlastní konvence, jež definuje různé znaky pro jednoduché formátování písma. Je to ale vlastnost, kterou si definují pouze některé programy a může se i mezi nimi navzájem lišit. Pro takovéto formátování neexistuje žádný standard. Oproti ostatním formátům má TXT jednu velkou výhodu. Je čitelný téměř na jakémkoli zařízení a to i v případě, že je v dokumentu zahrnut nějaký způsob nestandardního formátování.

Převod do tohoto formátu je také jednoduchý, stačí pouze vypustit všechny formátovací informace a vypsát jen samotný text.

### 2.1.9 Srovnání jednotlivých formátů

	Formátování písma	Odstavce	Nadpisy	Tabulky	Obrázky	Hypertextové odkazy	UTF-8
PalmDoc	ne	ne	ne	ne	ne	ne	ne
Mobipocket	ano	ano	ano	ano	ano	ano	ano
FB2	ano	ano	ano	ano	ano	ano	ano
Open E-Book	ano	ano	ano	ano	ano	ano	ano
EPUB	ano	ano	ano	ano	ano	ano	ano
Plucker	ano	ano	ano	ano	ano	ano	ano
iSilo	ano	ano	ano	ano	ano	ano	ano
Text	ne	ano	ne	ne	ne	ne	ano

Tabulka 2: Vlastnosti jednotlivých formátů

Nejmenší podporu pro jakékoli formátování, či vkládání obrázků, tabulek a dalších věcí má čistě textový formát. Tyto soubory nejsou nijak komprimovány ani kódovány. Jejich obsahem je pouze vlastní text. Veškeré tyto vlastnosti lze označit za jeho nevýhody, avšak právě ty z něj dělají formát, který je nejsnazší pro jakýkoli převod. Také je čitelný téměř na jakémkoli zařízení s jakýmkoli programovým vybavením

Přesným opakem v rámci snadnosti implementace jsou formáty PalmDoc a Mobipocket. Podle údajů v tabulce 2 by bylo možno říci, že PalmDoc se velice podobá TXT formátu. Podobnost je zde však pouze v tom, jaké má PalmDoc základní vlastnosti. Ty jsou však na rozdíl od TXT často dále rozšiřovány. Hlavní rozdíl spočívá také v tom, že PalmDoc, stejně jako Mobipocket jsou zapisovány do bitových kontejnerů. Tato vlastnost je dělá velice složitými pro jakoukoli implementaci.

Komprimace ani datových kontejnerů není využito u formátu FB2. Již díky tomu se jeví jako vhodný pro snadnou implementaci konvertoru. Jeho další vlastnosti a systém formátování je velice podobný samotnému HTML formátu a jeho struktura a podporované prvky nejvíce

odpovídají skutečné knize. Údaje pro tento formát obsažené v tabulce jsou vztaženy k verzi 2.2. Nižší verze ještě nemusí všechny tyto vlastnosti obsahovat.

Formát Open E-Book je v dnešní době zastaralý a místo něj se využívá EPUB. V jeho případě se jedná o komprimované XHTML soubory, díky čemuž se velice podobá formátu Plucker. V tomto případě jsou ale HTML soubory převedeny do binární podoby. Oba formáty podporují veškeré prvky používané v HTML dokumentech. Samotný EPUB používá ZIP komprimaci, čímž se stává nástrojem použitelným na většině zařízení.

Jediným uzavřeným formátem ze všech uvedených je iSilo. Již tato jediná vlastnost ho dělá velice nevhodným pro jakoukoli implementaci konvertoru. Avšak jinak je to jediný formát s možností značného zabezpečení a kontroly stárí obsahu.

## 2.2 Čtečky elektronických knih

Většina programů, jež jsou určeny pro čtení různých formátů elektronických knih, je implementováno pro běžné osobní počítače. Co do četnosti jsou v pořadí druhými PDA. Na běžné mobilní telefony, které jsou mezi uživateli nejrozšířenější, existuje paradoxně programů nejméně. Je to dáno hlavně omezenými možnostmi, plynoucími například z rozměrů jejich displeje. Ty jsou oproti stolním počítačům velice malé a jsou většinou menší než u PDA. Také platforma pro vývoj těchto programů má jen omezené vlastnosti a možnosti.

Programů ke čtení elektronických knih je k nalezení velké množství, většinou jsou však velice podobné a liší se třeba pouze v jedné jediné funkci a ve vzhledu či stylu reprezentace knih. Některé z nich dovedou přímo v rámci aplikace vyhledávat a stahovat knihy z Internetu. Všechny však podporují vyhledávání v souborovém systému zařízení.

### 2.2.1 FBReader a FBReaderJ

Nejznámějším ze všech programů pro čtení elektronických knih je právě FBReader. Svůj věhlas a známost si vydobyl tím, že je dostupný pro více platforem a také podporou mnoha formátů a jejich vlastností. Program je zcela zdarma.

Samotný FBReader je použitelný na běžných počítačích s operačními systémy Windows a Linux. Dále však i na tabletech či některých smartphonech.

Podporuje většinu nejrozšířenějších formátů elektronických knih. Patří mezi ně ePub, Fiction Book 2, Plucker a Mobipocket. Také dokáže číst knihy komprimované do balíčků různými

komprimačními technikami. Pokud je v takovémto balíčku více knih, dokáže jednotlivé z nich rozlišit.

FBReader vytváří z jednotlivých knih jakousi knihovnu, pomocí níž je velice usnadněna orientace a přechod mezi knihami. Dokáže také vytvářet obsah pro každou knihu v závislosti na její struktuře. Podporuje hypertextové odkazy i formátování. Umožňuje vyhledávání v textu nebo listování mezi jednotlivými stránkami. Při spuštění programu je vždy načtena poslední otevřená kniha a to přímo na pozici, v jaké byla před ukončením.

Největší nevýhodou tohoto programu je nemožnost zobrazit zbytek knihy v případě, že se v její struktuře vyskytne nějaká chyba. Potom je kniha zobrazena jen po tuto chybu. V nynější verzi také nejsou podporovány záložky. Jsou však plánovány k doplnění v další verzi, stejně jako další, zatím neurčené, formáty.

FBReaderJ je verze určená pro platformu Android napsaný v jazyce Java. Obsahuje většinu funkcí původního FBReadru, je zde však omezení v menším počtu podporovaných formátů. Umožňuje pouze čtení knih ve FictionBook 2, OEB a ePub.

### 2.2.2 Haali Reader

Je to aplikace primárně určená pro Pocket PC zařízení. Slouží ke čtení knih ve formátech PalmDoc, MobiPocket a dokáže otevřít také XML soubory.

Stejně jako FBReader i tento program otvírá vždy poslední čtenou knihu na místě, kde byla ukončena četba. Tím však podobnost mezi těmito čtečkami nekončí. I zde je totiž podporováno různé formátování, kapitoly, obsahy a opět je umožněno vyhledávání v textu. Aplikace je ruské výroby a je také zdarma.

### 2.2.3 ReadManiac

Jedná se o program na čtení elektronických knih vytvořený v JavaME převážně pro mobilní telefony značky Siemens, na ostatních není zaručená bezchybná funkčnost. Původní verze byly placené, ale nynější jsou již zdarma.

Program umožňuje vytvářet záložky, vyhledávat v textu a měnit velikost písma. Velikost jednotlivých knih je dána pouze velikostí fyzické paměti zařízení. Vyhledávání knih je možné nejen přímo v souborovém systému zařízení, ale také stahování knih z Internetu. Zvládá čtení formátu ZIP (může obsahovat kterýkoli z podporovaných formátů), PDB, PCR a TXT.



### 2.2.4 MobiPocket Reader

Je to další ze zástupců čtecích programů pro mobilní telefony obsahující JavaME, existuje však i verze pro Pocket PC. Jeho hlavní nevýhodou je pouze podpora formátů MobiPocket a PalmDoc. To však vynahrazuje velkou možností formátování textu, obrázky a také funkčními hypertextovými odkazy. Umožňuje i vytváření záložek a obsahů.

Existuje i jistá nastavba pro tento program. Ta kromě stahování novinek z Internetu dokáže i převod HTML a TXT souborů do prc formátu. Celá aplikace včetně její nastavby je zcela zdarma. Je však určena jen pro Pocket PC.

### 2.2.5 iSilo

Program iSilo je program vytvořený pro platformu Windows Mobile. Primárně je určen pro čtení knih ve stejnojmenném formátu iSilo. Podporuje však i několik dalších formátů jako PDB, PRC a HTML. Kromě podpory různých druhů formátování umožňuje samotný program i toto ručně nastavovat. Tato čtecí aplikace je však placená.

## 2.3 Konvertory elektronických knih

Elektronické knihy většinou vznikají převodem čistého textu nebo jiného textového dokumentu do některého z formátů, jichž se u elektronických knih využívá. K tomu je ovšem třeba využít nějaký program, konvertor, jenž je schopný takovýto převod uskutečnit. Samotné konvertory nepatří mezi příliš rozšířené programy. Přesto existují a mnohdy jsou to neplacené aplikace. Nejčastěji se vyskytují v takové podobě, kdy slouží pro převod HTML stránek do některého z použitelných formátů pro knihy. Cílem je poskytnout uživatelům možnost stáhnout si Internetovou stránku a její obsah moci prohlížet i bez připojení k Internetu na jakémkoli zařízení podporující čtení elektronických knih.

Ať už jsou vstupní soubory ve stejném nebo jiném formátu, vždy se výstupy z jednotlivých konvertorů od sebe podstatně liší. Jejich implementace převážně spočívá v nahrazování formátovacích nebo bílých znaků znaky jinými. Vybíráním informací o daném dokumentu a jejich následném zápisu. V některých případech i z výstupního kódování či komprimování. Zařazení prvků a jejich přesná podoba vždy závisí na tom, do jakých formátů je převod realizován.

V případě, že je potřeba při převodu jisté znaky vyhledávat a nahrazovat jinými, mohou vznikat nejasnosti v tom, jaký znak je adekvátní za ten původní. Z toho pak plynou i rozdíly mezi výstupy z jednotlivých programů sloužících pro konverzi formátů.

Dalším rozdílem mezi aplikacemi jsou stejně jako u čtecích programů vlastnosti, jež jsou z daného formátu implementovány. Ne vždy to jsou úplně všechny, jež formát specifikuje.

Většina konvertorů je implementována pro běžné počítače. Tím jsou odstraněny problémy s nedostatečným výkonem a chybějícími knihovnamy na mobilních zařízeních. Je tím ovšem vynucen následný přenos souboru, jenž již je v požadovaném formátu, do mobilního zařízení. Ne vždy je však k dispozici počítač nebo případně propojení mezi ním a mobilním zařízením.

To je možné odstranit konvertorem, jenž je implementován přímo pro takovéto zařízení. Rošíření MobiPocket Readeru takovýto konvertor obsahuje, zvládá však pouze konverzi do PRC souborů.

## 3 Implementační prostředí pro vývoj konvertoru

Nejrozšířenějším typem mobilního zařízení jsou stále ještě běžné mobilní telefony a z jejich stálého vývoje vyplývá, že tomu tak ještě značnou dobu bude. Všechny v dnešní době vyráběné mobilní telefony v sobě mají obsaženu platformu Java. Konkrétně se jedná o JavaME (Java 2 Micro Edition), což je pouze výtažek z běžně používané platformy JavaSE (Java Standard Edition). Obsahuje pouze ty nejn nutnější a nejpotřebnější knihovny. K tomu jsou navíc přidány ještě některé navíc, které se v běžné Javě nevyskytují. Slouží většinou k účelům, jež jsou vlastní pouze právě mobilním zařízením.

### 3.1 Základní konfigurace

Pro mobilní telefony, což jsou výpočetní zařízení s omezenými výpočetními, zobrazovacími a paměťovými možnostmi, se využívá konfigurace CLDC (Connected Limited Device Configuration). Jsou v ní obsaženy všechny základní knihovny z běžné Javy. CLDC nijak neřeší vlastní komunikaci nebo interakce se samotným zařízením. Podle možností a počtu obsažených knihoven jsou dnes k dispozici dvě verze CLDC. Pro implementaci konvertoru bude využito starší verze CLDC 1.0, což bude pro naše potřeby dostačující a zaručí to i kompatibilitu na jakémkoli mobilním zařízení s podporou jakékoli verze CLDC.

MIDP je profilem, jehož se využívá u mobilních telefonů. Je to nástavba CLDC, přičemž přidává možnost práce se sítí, uživatelským rozhraním a také přístup k lokálním úložištím. I u tohoto profilu existuje více verzí a stejně jako u CLDC i zde nové verze rozšiřují starší. V případě implementace konvertoru bude využito novější verze MIDP 2.0.

### 3.2 Nestandardní knihovny

V případě potřeby je možno při vývoji aplikace využít i jiných knihoven, jež nejsou obsaženy v CLDC ani MIDP. Pro implementaci konvertoru bude třeba použít jednu takovou knihovnu. Jedná se konkrétně o File Connection API, jež je součástí JSR-75. V něm je obsaženo ještě PIM API. To slouží například pro přístup ke kontaktům, událostem, poznámkám a kalendáři. Pro konvertor je však tato část JSR-75 nepotřebná. Využito bude pouze File Connection API, díky němuž bude možno přistupovat k souborům uloženým v paměťových prvcích mobilního zařízení (paměť telefonu, paměťová karta). Toto API obsahuje metody pro připojení k těmto souborům, jejich vytváření, mazání, zápis a čtení z nich.

Tato knihovna se využívá hlavně v zařízeních, kde je umožněno využívat paměťových karet. V dnešní době toto kritérium splňují již téměř všechny prodávané mobilní telefony

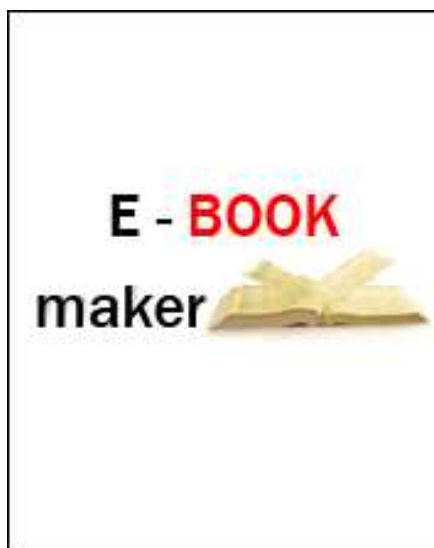
S používáním JSR-75 se pojí i otázka bezpečnosti dat uživatele. Aplikace se rozlišují na důvěryhodné a nedůvěryhodné. Jejich rozdělení je dáno tím, jestli užívají některý z podpisových certifikátů, jež jsou obsaženy v paměti mobilního zařízení nebo na SIM kartě. Pokud je aplikace shledána nedůvěryhodnou, podléhá veškerý přístup, zápis i čtení uživatelských dat kontrole samotného uživatele. Ten je při každé takovéto činnosti dotázán, zda si přeje, aby daná činnost proběhla či nikoli. Uživatel většinou nemá žádnou možnost jak tomuto předejít. V případě, že je aplikace podepsána pomocí platného certifikátu a mobilní zařízení ji shledá za důvěryhodnou, má uživatel možnost neustálému dotazování předejít a v zabezpečení programu nastavit, aby se aplikace nedotazovala nebo aby se dotázala pouze jednou.

## 4 Analýza programu

### 4.1 Základní popis programu

Aplikace bude na základě parametrů stanovených uživatelem převádět HTML soubory do několika formátů elektronických knih. Výstupní soubor v daném formátu bude ukládán do stejné složky jako původní převáděný dokument.

Při spuštění programu se uživateli na 1,5s zobrazí logo programu (viz obrázek 5). Následně bude uživateli rovnou zobrazeno převodní menu. V něm bude mít možnost zvolit pomocí prohlížeče, na nějž bude zvyklý i z jiných aplikací na daném zařízení, procházet souborový systém svého telefonu a zvolit soubor, který bude chtít převést. Aby nedošlo k tomu, že zvolí nepodporovaný formát vstupního souboru, budou mu zobrazovány pouze HTML dokumenty. Další možností uživatele bude zvolit si název výsledného souboru a také jeho formát. Na základě toho, jaký si vybere výstupní formát, mu bude automaticky doplněna přípona ke zvolenému názvu souboru.



Obrázek 5: Úvodní obrazovka s logem programu


Pokud to bude daný výstupní formát podporovat obrázky, bude uživatel dotázán před samotným průběhem převodu, jestli chce převod uskutečnit i s nimi. Při převodu bude uživatel animací informován o tom, že převod stále ještě probíhá. V jeho průběhu bude moci celý převod zastavit

a zrušit. Pokud tak neučiní, bude po převedení informován, jestli převod proběhl korektně nebo zda se při něm vyskytla nějaká chyba a nebyl uskutečněn nebo dokončen.

## 4.2 Uživatelské rozhraní

Uživatelské rozhraní celého programu bude snadno použitelné, všechny potřebné prvky pro nastavení a převod budou snadno nalezitelné. Veškeré nastavení se bude nacházet na jediném obrazovce. Výběr vstupního souboru a zápis jména nového souboru bude řešen v prvcích, jejichž vzhled bude telefon generovat dle svého vlastního nastavení vzhledu, tudíž budou uživateli důvěrně známé a velice snadno pro něj použitelné.

Pohyb mezi jednotlivými položkami nastavování bude zajištěn pomocí směrových tlačítek sloužících pro posun nahoru a dolů. Formát pro převod bude vybírán jednoduše tlačítky pro posun do stran. Vždy, když bude zapotřebí využít softwarových tlačítek, tak budou zřetelně označeny popisem. Při vyvolání informačního menu bude pohyb v něm opět pomocí směrových tlačítek a to logicky podle směru pohybu.

<b>E - BOOK</b> <b>maker</b> 	
<b>Původní soubor:</b> <input type="text"/>	
<b>Nový soubor:</b> <input type="text"/>	<b>Formát:</b> <input type="button" value="◀ TXT ▶"/>
<b>Další</b>	<b>Převést</b>

Obrázek 6: Obrazovka pro nastavování parametrů převodu

```
graph TD
    Start(( )) --> Načtení_znaku1([Načtení znaku])
    Načtení_znaku1 --> D1{ }
    D1 -- "[Znak '<']" --> End(( ))
    D1 -- "[Jiný znak]" --> D2{ }
    D2 -- "[Znak '<']" --> Načtení_znaku2([Načtení znaku])
    D2 -- "[Jiný znak]" --> Zápis([Zápis])
    Zápis --> Načtení_znaku1
    Načtení_znaku2 --> D3{ }
    D3 -- "[Znaky '>','='','']" --> Porovnávání([Porovnávání])
    D3 -- "[Jiný znak]" --> Tvorba_tagu([Tvorba tagu])
    Tvorba_tagu --> Načtení_znaku2
    Porovnávání --> Rozpoznávání([Rozpoznávání])
    Rozpoznávání --> Zápis
```

Jednotlivé znaky budou načítány postupně a vždy se prvně ověří, jestli načtený znak není ukončovacím symbolem souboru. Pokud ne, tak je dále ověřeno, zda se nejedná o počáteční znak HTML tagu. V případě, že se jedná o jiný znak, je tento symbol pouze přepsán na výstup a načte se další znak. Jakmile program nalezne znak začátku HTML tagu, je načten další symbol a podle jeho významu se buď skládá HTML tag, nebo se přistoupí k rozlišení tagu již složeného. To se skládá z části porovnávání, kdy je nalezený tag porovnávám s tagy v tabulce, a z části rozpoznávání. V ní se již konkrétně rozliší, o jaký HTML tag se jedná a jaká je příslušná reakce na něj. Celý postup je znázorněn na obrázku 7.

Při tvorbě HTML stránek se mnohdy kvůli přehlednosti kódu využívá značného počtu mezer, tabulátorů a odřádkování (viz. obrázek 8). Pokud by se nezavedl prostředek na jeho omezení, zanesly by se bílé znaky i do výstupního souboru. Z toho důvodu se budou kontrolovat počty těchto znaků jdoucích za sebou a v případě překročení limitu nebudou zapisovány do výstupního souboru.

```
<body>
  <div id="hlavni">
    <div id="menu">
      <table>
        <tr>
          <td width="150px" align="center" ><a href="index.html" ><b>Úvod</b></a></td>
          <td width="150px" align="center" ><a href="ja.html" ><b>O mně</b></a></td>
          <td width="150px" align="center" ><a href="rozvrh.html" ><b>Hůj rozvrh</b></a></td>
          <td width="150px" align="center" ><a href="odkazy.html" ><b>Odkazy</b></a></td>
        </tr>
      </table>
    </div>
    <div id="zpravy">
      <h3>Jmenuji se Zdeněk Vjater. Narodil jsem se roku 1988 v Ostravě a od té doby žiji
      stále zde. Přesněji řečeno v Ostraně-Zábřehu. Úspěšně jsem odmaturoval tehdy ještě
      na Střední škole Telekomunikační v oboru Digitální telekomunikační technika. Nyní
      studuji na Vysoké škole Báňské obor Mobilní technologie.</h3>
    </div>
  </div>
</body>
```

Dvojitě odřádkování způsobené snahou o přehlednost kódu

Mezery způsobené snahou o přehlednost

Obrázek 8: HTML kód a bílé znaky

Pokud by v průběhu došlo k nějaké chybě, kvůli níž by nebylo možno pokračovat, bude muset konvertor uzavřít všechna aktivní spojení se soubory a ukončit převod.



## 5 Implementace programu

### 5.1 Grafické rozhraní

Celé grafické rozhraní kromě dvou prvků je realizováno pomocí třídy Canvas. Jedinými prvky, které nejsou obsaženy ve třídě vytvořené pro účel komunikace s uživatelem, je pole pro vepsání textu a souborový prohlížeč. Propojení těchto prvků s implementovaným uživatelským rozhraním je řešeno přes spouštěcí třídu, v níž dochází k přepínání mezi nimi.

Veškeré přepínání a změny obrazovek jsou realizovány pomocí číselných proměnných. Ty se mění v závislosti na interakcích s uživatelem. Například podle jeho pohybu v menu či mezi prvky pro nastavování převodu. Teprve při dalším vykreslování jsou veškeré tyto proměnné vyhodnoceny a zvolí se odpovídající zobrazení.

Většina textových popisů v grafickém rozhraní je řešena obrázky, čímž dochází k nežádoucímu zvyšování celkové velikosti aplikace. Avšak zabrání se tím problémům s různými velikostmi písma na různých mobilních telefonech. V případě využívání obyčejného textu by mohlo docházet k situacím, při kterých by část jednoho textu překrývala jiný grafický prvek, nebo by se vůbec nevešla do vykreslované oblasti.

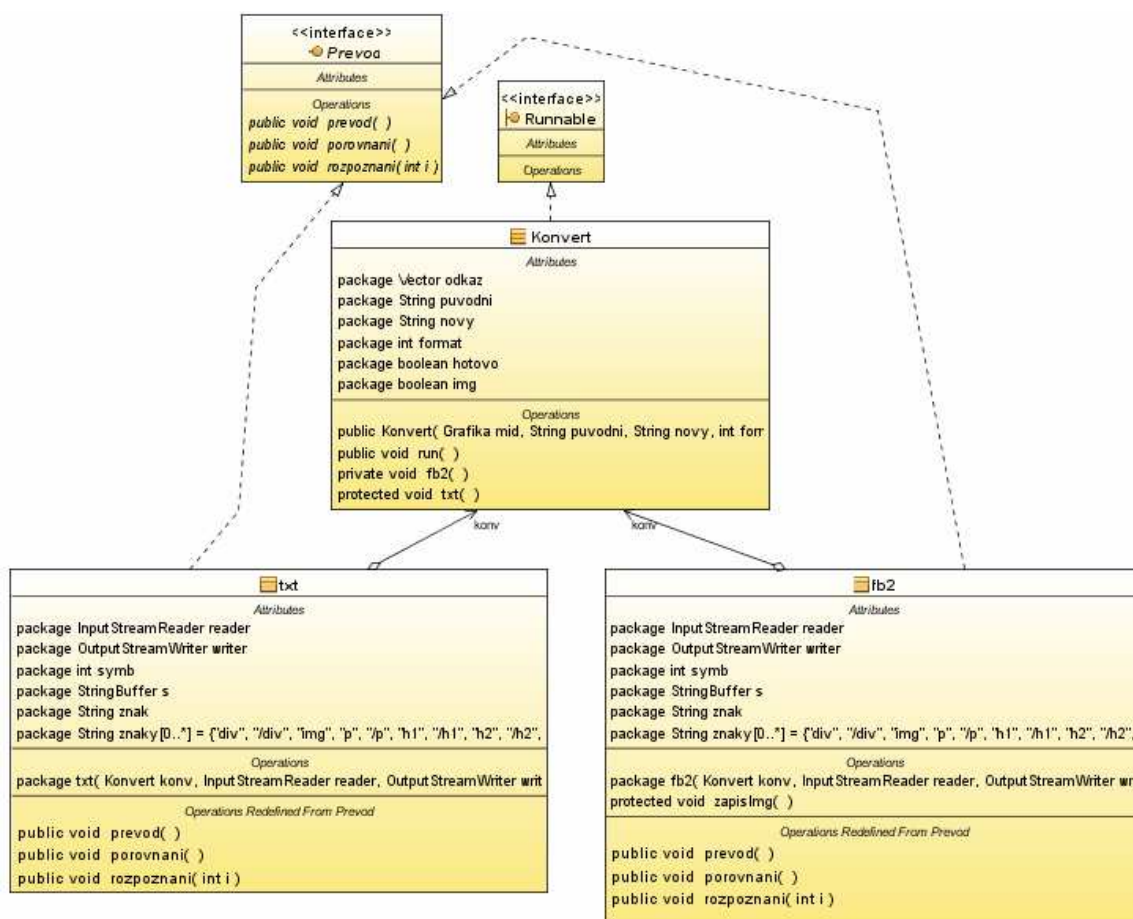
Při realizaci textových popisů musela být udělána jedna výjimka. V nastavení převodu jsou rámce, kde se má vypsát uživatelem zvolený vstupní soubor a název souboru výstupního. Pokud by se i zde mělo využívat vykreslování textu obrázky, bylo by nutno vytvořit celou abecedu znaků a zhotovit rozhodovací mechanismus, jenž by název souborů převedl na tuto abecedu. Nebo využít některou z existujících knihoven. Z toho důvodu je zde použit obyčejný text. Při spuštění grafického rozhraní je vždy zjištěna výška generovaného textu a podle ní je určena i výška rámců, do nichž se má tento text vpisovat. Zabrání se tím přesahování textu mimo vymezenou oblast.

### 5.2 Příprava převodu

Na obrázku 9 je zjednodušený třídní diagram části programu, která slouží pro převod formátů. Z důvodu přehlednosti jsou z něj vynechány některé proměnné v samotných třídách realizujících převod. Tyto proměnné většinou slouží k nastavování vnitřních stavů převodní třídy nebo k uchovávání dočasných dat, jež je využito ve více metodách.

Společná část *Konvert* je vytvořena a spouštěna jako nové vlákno vždy po vyžádání převodu. Proběhne zde rozhodnutí o tom, do jakého formátu se má převádět vstupní soubor. Na základě

toho je vybrána daná metoda. Ta je implementována pro každý formát vlastní. Ve svém základu se shodují. Vždy připraví všechny požadované spojení a vytvoří prvky pro čtení a zápis do souborů. Teprve až je vše připraveno vytvoří se třída pro převod do daného formátu a zavolá se její metoda řídící převod.



Obrázek 9: třídní diagram převodní části

V metodě pro FB2 je implementováno i řízení přechodu mezi odkazovanými soubory. Celý princip spočívá v tom, že jsou všechny odkazy na soubory existující na místním úložišti přidávány do seznamu, na jehož základě je postupně volán převod jednotlivých souborů. Před každým převodem dalšího souboru zahrnutého v jednom výstupním dokumentu je přidána značka pro odkaz na tuto pozici. Jakmile jsou převedeny všechny soubory obsažené v seznamu, je zahájen případný převod obrázků. Zde si již přepínání mezi jednotlivými soubory řeší sama třída pro převod do FB2 formátu. Po jejím provedení je zapsáno ukončení FB2 dokumentu.

Zápis odkazovaných souborů by se mohl dostat do nekonečné smyčky, pokud by na sebe navzájem odkazovaly dva soubory. Z tohoto důvodu je před přidáváním do seznamu odkazů vždy provedena kontrola, jestli zde již tento odkaz není zapsán. Pokud by tomu tak bylo, bude sice ve výstupním souboru vytvořen odkaz, ale již se neprovádí přidání do seznamu. Zabrání se tím nejen zacyklení převodu, ale také opakování stejných částí ve výstupním souboru. Stejného principu je využito i v případě obrázků. Zde by však nedocházelo k nežádoucímu opakování nebo vytvoření smyčky, ale k nežádoucímu narůstání výsledné velikosti výstupního souboru a také k prodloužení času potřebného k převodu. Bylo by naprosto zbytečné a neefektivní převádět stejný obrázek dvakrát.

### 5.3 Realizace převodu

Při implementaci konvertoru do jakéhokoli výstupního formátu, byly první nadefinovány HTML tagy, jež měly být při samotném převodu rozlišovány. V případě, že by byl zaznamenán tag, ale nebyl by nalezen v tabulce, je pouze dočten a na výstupu se nijak neprojeví. Při jejich výběru byl brán zřetel hlavně na jejich funkci a vliv na výstupní dokument. Každý formát má svou vlastní sadu. Důvodem je možnost rozšířit konverzi jednoho formátu o některý nový tag bez toho, aby bylo nutno zasahovat do všech implementovaných formátů. Musela by se totiž změnit nejen jejich tabulka, ale také přidat reakci na něj.

0	<div>	15	 
1	</div>	16	<a>
2	<img>	17	</a>
3	<p>	18	<form>
4	</p>	19	</form>
5	<h1>	20	<ul>
6	</h1>	21	</ul>
7	<h2>	22	<li>
8	</h2>	23	</li>
9	<table>	24	<ol>
10	</table>	25	</ol>
11	<td>	26	<b>
12	</td>	27	</b>
13	<tr>	28	<i>
14	</tr>	29	</i>

Tabulka 3: Základní tabulka HTML znaků

Prvním formátem, jenž byl vybrán pro implementaci je textový formát TXT. V jeho případě se může tabulka pro rozeznávání formátování až příliš rozsáhlá, ale kromě některých tagů jsou ostatní nutné pro rozlišování míst, kde je potřeba vložit odřádkování. Nic jiného se do výstupního souboru nepřidává. Pouze se přepisuje vlastní text bez formátovacích tagů a v případě potřeby je vloženo zmíněné řádkování. Kromě toho se už pouze kontrolují nadbytečné mezery a prázdné řádky.

Druhým formátem je FB2. Zde již je pokročilejší reakce po rozeznání jednotlivých formátovacích tagů. Ve většině případů jsou nahrazeny tagy definovanými v samotném FB2 formátu. Jednotlivé sekce jsou děleny podle stránek. Všechny odkazy na soubory existující v souborovém systému na odkazovaném místě jsou ukládány do seznamu a po převedení aktuální stránky jsou postupně také převedeny. Pokud je zvolen převod s obrázky, probíhá při jejich nalezení stejná kontrola jako při odkazech na další stránky. Veškeré obrázky jsou do výstupního souboru zapsány až úplně na závěr. Naopak na začátku souboru se po přečtení hlavičky převáděného souboru zapíše hlavička pro FB2 soubor. Konkrétně se do ní zapisuje autor a titul. Pokud by nebyla HTML hlavička vyplněna, budou tyto položky ve výsledné elektronické knize prázdné.

Třetím formátem měl být ePub. Zde bylo dostačující pozměnit hlavičku původního HTML dokumentu a vytvořit soubory obsahující údaje o struktuře elektronické knihy a také soubor obsahující její metadata. Všechny ostatní věci, jako jsou obrázky či dokonce CSS soubor by stačilo pouze překopírovat do jedné složky. Zde se však vyskytnul problém s omezením, jež plyne z využití JavaME. Kniha ve formátu ePub je totiž pouze ZIP balíček se změněnou příponou. JavaSE obsahuje knihovny pracující se ZIP soubory, včetně jejich rozbalování a komprimování. V JavaME ale obsaženy nejsou. Existují sice různé uživatelské implementace realizující tuto knihovnu, avšak většinou se jedná pouze o rozbalování souborů anebo v případě, že už se jedná i o jejich komprimaci, tak pomocí nich lze komprimovat pouze jeden soubor. Pro využití k zabalení ePub knihy jsou nepoužitelné, protože u ní je zapotřebí zabalit celou adresářovou strukturu této knihy. Realizace převodu pro tento formát se kvůli tomu nezdařila.

## 5.4 Přidání nového formátu

V konvertoru je nyní implementován převod pouze do dvou formátů. Je zde však poměrně snadná možnost dalšího budoucího rozšiřování. V případě takového kroku je v grafickém rozhraní nutno inkrementovat proměnnou *typM* o hodnotu rovnu počtu přidávaných formátů. Pak je nutno ještě přidat obrázek odpovídající názvu daného formátu v menu. Ve třídě *Konvert* je pak nutné dodat do přepínače možnost dalšího formátu a podle ostatních stávajících metod upravit metodu pro přípravu k převodu.

Samotná převodní třída by měla implementovat připravené rozhraní *Převod*, v němž je nadefinována struktura pro vlastní převod souborů. Jelikož bude při rozšiřování nutno udělat zásah i do přípravné třídy *Konvert*, má rozhraní funkci spíše orientační. Je možno novou převodní třídu vytvořit jakkoli, ale programátor by v takovémto případě musel přizpůsobit i svou metodu v přípravné třídě. Nebylo by možno už jen upravit metodu určenou k převodu do jiného formátu. Pokud dodrží stávající strukturu přípravných metod a následně bude jím nově vytvořená třída implementovat zmíněné rozhraní, bude opět celý převod souboru dodržovat navržené schéma znázorněné na obrázku 7. Pro jakýkoli formát bude moci využít i stejné realizace některých těchto metod. Při rozšiřování bude nutno si vytvořit vlastní tabulku HTML tagů.

## 6 Testování programu

Testování aplikace proběhlo na třech mobilních telefonech od dvou různých výrobců. Všechna tato zařízení měla dostatečnou konfiguraci Java profilů pro potřeby implementovaného konvertoru. Testovací zařízení byly od společnosti Nokia a Sony Ericsson. V případě Nokia se jednalo o typy 3110c a 5330. Jediným zástupcem od Sony Ericsson byl typ W810i.



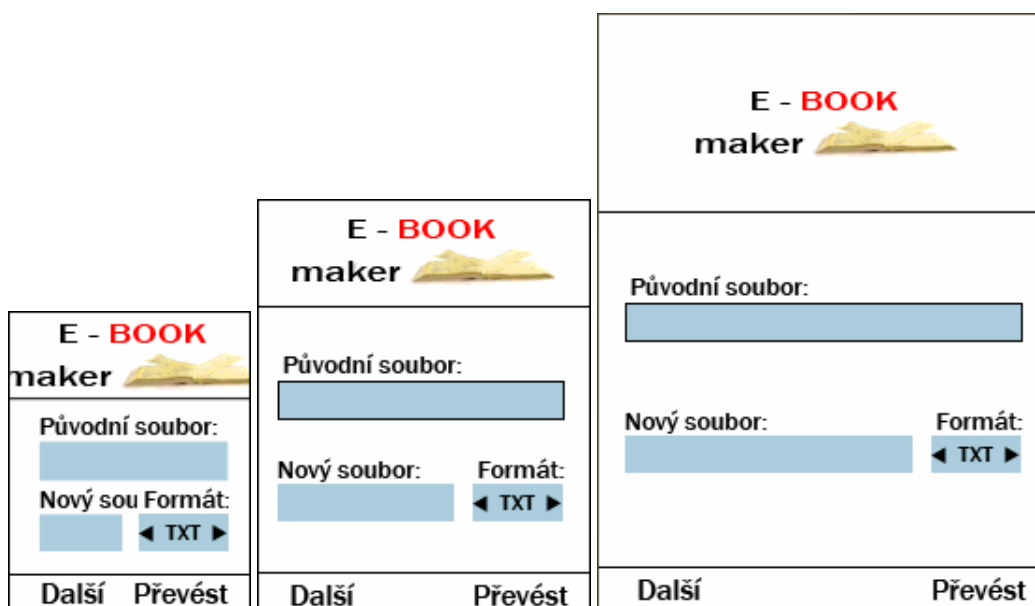
Obrázek 10: Nokia 3110c, Sony Ericsson W810i a Nokia 5330 Express Music

Aplikace má grafické rozhraní vytvořené pro minimální velikost displeje 176x220 bodů. Úmyslně byly vybrány telefony s různými rozměry zobrazovací jednotky, přičemž byly zvoleny možnosti větší, menší a rovny minimálním parametrům. Zařízení s menším displejem bylo vybráno proto, aby se dokázala funkčnost a možnost práce v tomto programu i s chybně zobrazovaným rozložením prvků na obrazovce.

Hlavním rozdílem v chování na jednotlivých testovaných mobilních telefonech je doba potřebná k převodu. Tento rozdíl je patrný již při pouhém převodu krátkého HTML souboru (velikost souboru 3kB) do formátu TXT. Telefon W810i tento úkon s pomocí implementovaného konvertoru provedl téměř okamžitě. Čas potřebný pro převod byl nižší než 0,5s. U Nokie 3110c byl tento čas větší než 1s. V případě testování na rozsahově menších souborech, porovnávaný soubor měl velikost 14kB, se tento údaj pohyboval i v rozmezí 5s. Minimální čas na stejném souboru byl pod 1s. Je to dáno nejen výkonem samotného mobilního telefonu, ale také tím, kterému vlákně programu přiřadí jakou prioritu a s ní spojený výkon. U telefonů firmy Nokia

byla animace při převodu plynulá s rychlostí překreslování odpovídající její implementaci. V případě Sony Ericsson bylo překreslování snímku značně pomalejší, ale samotný převod naopak mnohem rychlejší.

Čas zpracovávání souboru na jednom zařízení byl v případě převodu do formátu FB2 dán hlavně počtem odkazů na jiné soubory či případně obrázky, nezáleží pouze na tom, kolik takových souborů existuje. Nejdelší dobu z celého převodu totiž v takovýchto případech zabere právě ověřování, jestli jde o odkazy na existující soubory a jestli je tedy potřeba je následně také převést. Při jejich velkém počtu dochází ke značnému navýšení času. Nejnáročnější na převod by se tím staly graficky náročné stránky s rozsáhlým větvením.



Obrázek 11: Zobrazování při rozměrech displeje 128x160, 178x220 a 240x320 bodů

Program je kromě omezeného zobrazování na telefonu Nokia 3110c plně funkční na všech testovaných zařízeních a podává shodné výstupní soubory. Tyto soubory jsou čitelné v jakýchkoli programech sloužících ke čtení elektronických knih ve formátech, jež umí vytvářet implementovaný konvertor (TXT a FB2). U formátu FB2 je v této implementaci zahrnut i převod tabulek. Ty však většina čtecích programů neumí správně zobrazit. Druhá alternativa byla pouze se pokusit od sebe jednotlivé části tabulky oddělit pomocí formátování do odstavců. U předních zástupců čtecích programů však dochází ke stálému vývoji, takže se dá předpokládat, že v některé z dalších verzí již podpora tabulek bude. V některých dalších čtecích programech může nastat problém s odkazy. Implementovaný konvertor je vytváří, ale opět v mnohých čtecích aplikacích chybí jejich podpora. Zde to však nebude mít žádný vliv na zobrazování.

Ve srovnání s jedním ze zástupců konvertorů HTML stránek do formátu FB2 dosahuje konvertor pro mobilní zařízení shodných výsledků při použití malých vstupních souborů. V případě souborů větších (nad 500kB) mobilní zařízení potřebují mnohem delší dobu k převodu. Tato doba se pohybuje řádově až v minutách a je závislá i na konkrétním mobilním telefonu (mezi jednotlivými telefony jsou rozdíly mezi časy potřebnými na převod opět řádově v minutách). Počítačová aplikace si uchovává dobu převodu i přes zvyšování velikosti souboru řádově v sekundách.

Srovnávaný program Any to FB2 má sice mnohem více možností nastavení pro výslednou knihu a dokáže stejně jako implementovaný konvertor vytvářet knihu z více propojených stránek, ale nevytváří funkční vnitřní odkazy v dokumentu. Jeho přední výhodou vyplývající ze srovnávání spočívá v možnosti procházet i odkazy nacházející se mimo místní úložiště dat, může tedy přistupovat i k Internetu a další stránky či obrázky stahovat. Výsledný dokument se stejným nastavením, jaké je možné dosáhnout v konvertoru pro mobilní zařízení, byl v podání programu Any to FB2 velikostně téměř shodný.

Velkým omezením pro implementovaný konvertor může být v případě, že nebude podepsán platným certifikátem, časté dotazování programu na možnost čtení či zápisu dat. To se děje při jakékoli činnosti, při níž je nutno přistupovat k souborům nacházejícím se v mobilním zařízení. V průběhu výběru vstupního souboru je četnost takovýchto dotazů dána převážně hloubkou umístění daného souboru v adresářové struktuře. Při samotné konverzi do TXT je uživatel dotázán pouze dvakrát. V okamžiku, kdy program otevírá spojení se vstupním a výstupním souborem. Jinak tomu však již je u FB2. Zde je četnost dotazování dána počtem odkazů na jiné soubory. To je možné omezit výběrem převodu bez obrázků.



## 7 Závěr

V průběhu této práce vznikl funkční konvertor HTML stránek do formátu elektronických knih. Ve finální verzi převádí do formátu TXT a FB2. V průběhu implementace byl pokus přidat i převod do ePub formátu. Zde se však projevilo omezení platformy JavaME, když bylo zapotřebí výsledné soubory komprimovat do ZIP archivu. Knihovny, jež jsou k tomuto účelu v JavaSE běžně dostupné, nejsou součástí námi zvolené JavaME.

Při porovnání programu s jiným, pracujícím na běžném počítači, dosáhl konvertor shodných výsledků a také kvality převodu. Byl však implementován pouze pro offline převod HTML souborů, takže v něm není možnost připojit se k Internetu a převádět i stránky, na něž zde vedou odkazy. To by však bylo možno realizovat jednoduchou úpravou v nynější aplikaci. Stahování souborů by se však v případě rozsáhlejších stránek z důvodů zpoplatněného připojení mohlo stát velice nákladným.

Konvertor vznikl s určitým počtem formátů, do nichž lze vstupní soubory převádět. Tento počet lze však pomocí již připravených prvků v programu dále navýšit implementací převodníku do některého z dalších formátů popsaných v průběhu této práce.

Dalším do budoucna možným rozšířením programu je implementace konverze mezi jednotlivými formáty elektronických knih navzájem.

## 8 Literatura


- [1]Java.sun.com [online]. 2004 [cit. 2010-04-26]. Getting Started with the FileConnection APIs. Dostupné z WWW: <<http://developers.sun.com/mobility/apis/articles/fileconnection/>>.
- [2]Haali.su [online]. 2002 [cit. 2010-04-26]. FictionBook description. Dostupné z WWW: <[http://haali.su/pocketpc/FictionBook\\_description.html](http://haali.su/pocketpc/FictionBook_description.html)>.
- [3]FBReader.org [online]. 2009 [cit. 2010-04-26]. FBReader — e-book reader for Unix/Windows computers. Dostupné z WWW: <<http://www.fbreader.org/>>.
- [4]Wiki.mobileread.com [online]. 2009 [cit. 2010-04-26]. FB2. Dostupné z WWW: <<http://wiki.mobileread.com/wiki/FB2>>.
- [5]Fictionbook.org [online]. 2006 [cit. 2010-04-26]. FictionBook. Dostupné z WWW: <<http://www.fictionbook.org/index.php/Eng:FictionBook>>.
- [6][LD] Technika [online]. 2001 [cit. 2010-04-26]. Formáty elektronických knih. Dostupné z WWW: <<http://ld.johannesville.net/technika/02-formaty-elektronickych-knih>>.
- [7]ISilo™ [online]. 1999 [cit. 2010-04-26]. ISilo. Dostupné z WWW: <<http://www.isilo.com/support/index.htm>>.
- [8]PalmKnihy [online]. 2006 [cit. 2010-04-26]. Jak číst: Mobilní zařízení s podporou Javy. Dostupné z WWW: <<http://palmknihy.cz/www/index.php?PLATFORM=mobile&SW=Java>>

## 9 Seznam příloh

A	Uživatelská příručka.....	31
B	Konvertor.....	na CD
C	Programátorská dokumentace.....	na CD
D	Vstupní soubory pro testování programu.....	na CD
E	Výstupní soubory pro testování programu.....	na CD

## A Uživatelská příručka

Aplikace umožňuje převod HTML souborů do formátu elektronických knih na základě nastavených parametrů. Před převodem je možno pomocí standardního prohlížeče souborů zvolit dokument, jenž má být převeden. Následně název výstupního souboru a požadovaný výstupní formát.

<b>E - BOOK</b> <b>maker</b> 	
Původní soubor: <input type="text"/>	
Nový soubor: <input type="text"/>	Formát: <input type="button" value="◀ TXT ▶"/>
<b>Další</b>	<b>Převést</b>

Při nastavování původního souboru se pouze v klasickém prohlížeči vybere soubor, jenž má být převeden. Položka Nový soubor slouží k zápisu názvu nového souboru. Jeho přípona je doplněna automaticky podle aktuálně zvoleného formátu. Nový soubor bude po vytvoření uložen do stejného adresáře jako soubor převáděný.

Program je ovládán směrovými, softwarovými a potvrzovacími klávesami. Pro pohyb mezi jednotlivými položkami pro nastavování převodu slouží klávesy pro směr dolů a nahoru. Pro výběr polí pro název souborů je nutno stisknout potvrzovací klávesu. Po označení položky formát je možno změnit požadovaný výstupní formát klávesami pro směr vlevo a vpravo. Textové popisky ve spodní části obrazovky slouží k označení funkcí softwarových kláves.